

Summary of Phase P1 Results

Phase P1 consists of two tasks:

- [T1] Task T1: Analysis and design of finite wordlength implementations of linear time-invariant  $\delta$ -Systems.
- [T2] Task T3: 2-D and  $m$ -D  $\delta$ -system models.

This project is an extensive collaborative effort with Professor Peter H. Bauer, Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556, who is the principal investigator of

Grant No: N00014-94-1-0387;  
R&T Project Code: 3148509—01.

This report, and the research work indicated therein, were hence completed in cooperation with him.

Major part of task T1 was carried out at the University of Notre Dame by Dr. Peter H. Bauer while major part of task T3 was carried out at the University of Miami by Dr. Kamal Premaratne. Of course, the two principal investigators have been in constant contact throughout the work.

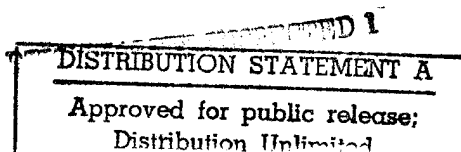
The following is a summary of the phase P1 results.

**Task T1: Analysis and Design of Finite Wordlength Implementations of Linear, Time-Invariant  $\delta$ -Systems**

The conclusions drawn from the work conducted for task T1 may be summarized as follows:

1. The Fixed-Point Arithmetic Case: When limit cycle performance is crucial, the  $q$ -operator implementation is preferable. The  $\delta$ -operator implementation is superior with regard to coefficient sensitivity issues.
2. The Floating-Point Arithmetic Case: Generally, the  $\delta$ -operator implementation outperforms its  $q$ -operator counterpart. In particular, in high-order and high-speed applications, the  $\delta$ -operator implementation is the best choice.

19950925 062



Prior to a more detailed exposition, first we provide qualitative justification for the above conclusion. The state equations of a  $\delta$ -operator system can be written as:

$$\begin{aligned}\delta[\mathbf{x}](n) &= A_\delta \mathbf{x}(n) + B_\delta \mathbf{u}(n); \\ q[\mathbf{x}](n) &= \mathbf{x}(n) + \Delta \cdot \delta[\mathbf{x}](n).\end{aligned}\tag{T1.1}$$

where  $\mathbf{x}$  and  $\mathbf{u}$  are the state and input vectors, respectively. Here,  $\Delta$  denote a positive real constant (typically, the sampling time). The symbol  $\delta[\cdot]$  denotes the  $\delta$ -operator, that is,

$$\delta[\mathbf{x}](n) = \frac{q[\mathbf{x}](n) - \mathbf{x}(n)}{\Delta} = \frac{q - 1}{\Delta} \mathbf{x}(n),\tag{T1.2}$$

and  $q[\cdot]$  denotes the usual  $q$ -operator, that is,

$$q[\mathbf{x}](n) = \mathbf{x}(n + 1).\tag{T1.3}$$

The corresponding formulation of (T1.1) in terms of the  $q$ -operator is

$$q[\mathbf{x}](n) = A_q \mathbf{x}(n) + B_q \mathbf{u}(n),\tag{T1.4}$$

where

$$A_q = I + \Delta \cdot A_\delta \iff A_\delta = \frac{A_q - I}{\Delta} \quad \text{and} \quad B_q = \Delta \cdot B_\delta \iff B_\delta = \frac{B_q}{\Delta}.\tag{T1.5}$$

Now, given  $\mathbf{x}$  and  $\mathbf{u}$ , both representations compute  $q[\mathbf{x}]$  with a certain accuracy. Consider the  $\delta$ -operator formulation in (T1.1). Here we encounter two errors:

1. The first is due to the computation of  $\delta[\mathbf{x}]$ , that is, the first equation in (T1.1). We will refer to this equation as the *intermediate equation*.
2. The second is due to the eventual computation of  $q[\mathbf{x}]$ , that is, the second equation in (T1.1). We will refer to this equation as the *update equation*.

Let us assume that the total error in computing  $q[\mathbf{x}]$  is mainly due to the intermediate equation in (T1.1) (rather than the update equation). Then, by choosing  $\Delta$  sufficiently small, the total error in computing  $q[\mathbf{x}]$  will be approximately the error created by the update equation which is small!. In this case, the  $\delta$ -operator representation has better finite wordlength properties than its  $q$ -operator counterpart in (T1.4).

If, however, the errors accumulated in the intermediate and the update equations in (T1.1) are comparable,  $q[\mathbf{x}]$  computed through the  $\delta$ -operator representation will show

approximately the same error as that computed through its  $q$ -operator counterpart assuming  $\Delta$  is sufficiently small. If  $\Delta$  is not sufficiently smaller than one, the  $\delta$ -operator representation will actually perform worse than the  $q$ -operator representation!

If the error introduced in the update equation is larger than that in the intermediate equation, the  $\delta$ -operator representation would consistently perform worse!! In reality, this case is very unlikely to occur.

Next, a more detailed exposition follows.

### T1.1 The Fixed-Point Arithmetic Case

We now discuss some of the results regarding the fixed-point (FXP) case. Here, our results in fact indicate that, in case limit cycle behavior is crucial, the  $\delta$ -operator representation is NOT suitable with this arithmetic scheme [1]. Such a case may occur when nonlinear systems are implemented through FXP  $\delta$ -operator based schemes.

*Zero-input limit cycles.* Independent of  $\Delta$ , zero-input limit cycles cannot be avoided in FXP  $\delta$ -implementations. This is easily explained as follows: If  $\Delta$  is chosen very small, the contribution from the intermediate equation being small (since  $\delta[x]$  is being multiplied by  $\Delta$ ), during the update equation,  $q[x]$  can be quantized to  $x$  creating a DC limit cycle, that is, an incorrect equilibrium point different from zero results. We emphasize that, most of the desirable properties of  $\delta$ -operator implementations are based on a small  $\Delta$ . We may also show that, if  $\Delta$  is chosen larger (this case is of course somewhat less important), DC limit cycles will still exist. Hence,  $\delta$ -operator representations cannot be implemented limit cycle free in FXP format! This fact is independent of the particular realization of the system.

*Deadband size.* Since  $\delta$ -systems cannot be implemented limit cycle free in FXP format, it is of interest to investigate the size of such limit cycles since, in certain situations, such small limit cycle amplitudes can be tolerated. It can be shown that, the magnitude of  $\Delta$  determines the magnitude of the limit cycle. The smaller the  $\Delta$ , the larger will be the deadband and hence the limit cycle magnitude. An approximate relationship regarding this is

$$\Delta \times \text{size of deadband} = 1, \quad (\text{T1.6})$$

where the size of deadband is measured in multiples of the quantization step size. Here, the deadband corresponds to that obtained by considering the quantization of  $\Delta \cdot \delta[x]$ . Therefore, the usual choice of a small  $\Delta$  creates a larger deadband!

Availability Codes	
Dist	Avail and/or Special
A-1	

*The input driven case.* Although the input driven case is not part of the originally proposed work, some interesting results have been obtained. For small values of  $\Delta$ , there exists a bounded input signal that does not allow control of the state trajectory. In other words, given sufficiently small  $\Delta$ , the state trajectory may not be influenced by such an input signal.

*The influence of the realization.* First, it was necessary to develop a suitable scheme to investigate the effect of realization on the presence or absence of limit cycles. In this direction, for the  $q$ -operator case, a computer-based exhaustive search algorithm that checks for limit cycles (DC and/or oscillatory) has been developed [5].

As discussed before, we have shown that, a stable linear time-invariant  $\delta$ -system cannot be implemented limit cycle free in FXP. The size of the deadband however also depends on the particular realization, that is, the structure of  $A_\delta$ . Given a system transfer function, there are forms which minimize this deadband size with respect to some appropriately chosen measure. For example, in order to minimize DC limit cycle amplitude, one may choose the normal form (in terms of  $A_\delta$ ) as a suitable candidate.

*The influence of quantization nonlinearity and its deadzone.* Since a larger deadzone implies larger DC limit cycle amplitudes, the use of quantizers with reduced, or even zero, deadzone was therefore proposed. In investigating first-order systems, by reducing the deadzone, it was found that, existence of DC limit cycles can indeed be reduced. Unfortunately, other oscillatory limit cycles will be created. This phenomenon is due to the increased gain exhibited towards small input signals by the quantizer.

*Scaling.* As discussed above, we have shown that, independent of either the form of  $A_\delta$  or the magnitude of  $\Delta$ , a FXP implemented  $\delta$ -system cannot be free of zero-input limit cycles. Hence, scaling cannot be offered as a possible solution.

### *T1.2 The Floating-Point Arithmetic Case*

The floating-point (FLP) implementation of  $\delta$ -systems is currently under investigation. The results obtained so far are very encouraging, and indicate that, quantization errors due to FLP arithmetic have a much smaller effect on the system behavior than in the FXP case. In fact, preliminary results show that, for  $\delta$ -systems of order three and higher, errors in computing  $q[\mathbf{x}]$  can be made significantly smaller than for the corresponding  $q$ -systems. This is because, for a FLP implementation of such a system, errors created through the intermediate equation are larger than those created through the update equation. As previously mentioned, in this situation,  $\delta$ -systems behave better than their  $q$ -operator

counterparts!

*Limit cycles.* In FLP arithmetic, a linearly stable time invariant system, under zero-input conditions, may exhibit four types of responses: A diverging response, an oscillatory periodic response of arbitrary magnitude, an oscillatory periodic response in underflow, or an asymptotically stable response. Only the last two response types are acceptable in practice. It is well known that, the last response type is in fact a very stringent requirement and is often not required in practice. Results so far obtained show that, when the requirements for a response in underflow are compared, the  $\delta$ -system requires less wordlength than its  $q$ -system counterpart! This advantage in fact grows with the order of the system!!

Once the system reaches underflow conditions, the  $\delta$ -system again exhibits DC limit cycles. However, if the exponent register is chosen sufficiently large, the amplitude of these oscillations can be made extremely small and hence, for all practical purposes, this problem is solved.

*Deadband size.* If the condition on the mantissa length that guarantees convergence into underflow is satisfied, then the deadband size will be very small. Hence, it can be neglected for all practical purposes. This assumes a properly chosen exponent register length since the exponent register length determines the dynamic range of underflow.

*The Influence of the Nonlinearity.* Unlike the FXP case, the characteristic of the nonlinearity has only a minor effect on the system behavior, significant differences being present only in underflow conditions

*The Underflow case.* In underflow, the  $\delta$ -system seems to behave worse than its  $q$ -operator counterpart. This is mainly due to the fact that, a FLP system in underflow essentially performs very similar to a FXP system. However, as mentioned above, if the dynamic range of underflow is chosen properly, the system behavior in underflow is of little practical interest.

*Block Floating-Point Arithmetic.* Even for the  $q$ -operator case, results regarding block FLP implementations are lacking. Hence, investigations regarding block FLP implementation of  $\delta$ -systems is in its early stages. In order to obtain a comparison between the two types of implementations, current research is geared towards obtaining results applicable for the  $q$ -operator case.

### T1.3 The Multi-Dimensional Case

The results on one-dimensional (1-D)  $\delta$ -operator implementations in FXP arithmetic directly carry over to the multi-dimensional ( $m$ -D) case. The existence of non-converging responses along the boundary of the causality region can easily be proven using the same type of argument used in the 1-D case. Consequently,  $\delta$ -operator based implementations of  $m$ -D systems cannot be implemented limit cycle free in FXP.

#### **Task T3: 2-D and $m$ -D $\delta$ -system models**

Discrete-time systems implemented using the  $\delta$ -operator, as is clear from the discussion above, exhibit superior finite wordlength properties with FLP arithmetic. In the case of FXP arithmetic, they still provide superior coefficient sensitivity. The development of 2-D and  $m$ -D models applicable for  $\delta$ -operator implementations was hence motivated with the expectation that these properties would still hold true.

The conclusions drawn from the work conducted for task T3 may be summarized as follows: Similar to the 1-D case, under FLP arithmetic, the  $\delta$ -operator implementation of 2-D and  $m$ -D discrete-time systems provides the best choice. Again, this is particularly true in high-order and high-speed applications.

*State-space models.* In Roesser local s.s. model of  $q$ -operator formulated 2-D discrete-time systems takes the form

$$\begin{aligned}
 \begin{bmatrix} q_h[\mathbf{x}^h](i, j) \\ q_v[\mathbf{x}^v](i, j) \end{bmatrix} &= \begin{bmatrix} A_q^{(1)} & A_q^{(2)} \\ A_q^{(3)} & A_q^{(4)} \end{bmatrix} \begin{bmatrix} \mathbf{x}^h(i, j) \\ \mathbf{x}^v(i, j) \end{bmatrix} + \begin{bmatrix} B_q^{(1)} \\ B_q^{(2)} \end{bmatrix} \mathbf{u}(i, j) \\
 &\doteq [A_q] \begin{bmatrix} \mathbf{x}^h(i, j) \\ \mathbf{x}^v(i, j) \end{bmatrix} + [B_q] \mathbf{u}(i, j); \\
 \mathbf{y}(i, j) &= [C_q^{(1)} \quad C_q^{(2)}] \begin{bmatrix} \mathbf{x}^h(i, j) \\ \mathbf{x}^v(i, j) \end{bmatrix} + [D_q] \mathbf{u}(i, j) \\
 &\doteq [C_q] \begin{bmatrix} \mathbf{x}^h(i, j) \\ \mathbf{x}^v(i, j) \end{bmatrix} + [D_q] \mathbf{u}(i, j),
 \end{aligned} \tag{T3.1}$$

where  $A_q^{(1)}$  is of size  $n_h \times n_h$ ,  $A_q^{(4)}$  is of size  $n_v \times n_v$ , etc. Also,  $q_h[\cdot]$  and  $q_v[\cdot]$  denote the horizontal and vertical shift operators, that is,

$$q_h[\mathbf{x}](i, j) = \mathbf{x}(i + 1, j) \quad \text{and} \quad q_v[\mathbf{x}](i, j) = \mathbf{x}(i, j + 1). \tag{T3.2}$$

To exploit the advantages of  $\delta$ -operator implementations, analogous to the 1-D case,

we define the operators

$$\begin{aligned}\delta_h[\mathbf{x}](i, j) &= \frac{\mathbf{x}(i+1, j) - \mathbf{x}(i, j)}{\Delta_h} = \frac{q_h[\mathbf{x}](i, j) - \mathbf{x}(i, j)}{\Delta_h}; \\ \delta_v[\mathbf{x}](i, j) &= \frac{\mathbf{x}(i, j+1) - \mathbf{x}(i, j)}{\Delta_v} = \frac{q_v[\mathbf{x}](i, j) - \mathbf{x}(i, j)}{\Delta_v},\end{aligned}\tag{T3.3}$$

where  $\Delta_h$  and  $\Delta_v$  are two positive real constants. The corresponding  $\delta$ -operator s.s. model may then be obtained as

$$\begin{aligned}\begin{bmatrix} \delta_h[\mathbf{x}^h](i, j) \\ \delta_v[\mathbf{x}^v](i, j) \end{bmatrix} &= \begin{bmatrix} A^{(1)} & A^{(2)} \\ A^{(3)} & A^{(4)} \end{bmatrix} \begin{bmatrix} \mathbf{x}^h(i, j) \\ \mathbf{x}^v(i, j) \end{bmatrix} + \begin{bmatrix} B^{(1)} \\ B^{(2)} \end{bmatrix} \mathbf{u}(i, j) \\ &\doteq [A] \begin{bmatrix} \mathbf{x}^h(i, j) \\ \mathbf{x}^v(i, j) \end{bmatrix} + [B] \mathbf{u}(i, j); \\ \mathbf{y}(i, j) &= [C^{(1)} \quad C^{(2)}] \begin{bmatrix} \mathbf{x}^h(i, j) \\ \mathbf{x}^v(i, j) \end{bmatrix} + [D] \mathbf{u}(i, j) \\ &\doteq [C] \begin{bmatrix} \mathbf{x}^h(i, j) \\ \mathbf{x}^v(i, j) \end{bmatrix} + [D] \mathbf{u}(i, j).\end{aligned}\tag{T3.4}$$

This is the 2-D version of the intermediate equation mentioned earlier. In addition, as for the 1-D case, we have the following update equations:

$$\begin{aligned}q_h[\mathbf{x}^h](i, j) &= \mathbf{x}^h(i, j) + \Delta_h \cdot \delta_h[\mathbf{x}^h](i, j); \\ q_v[\mathbf{x}^v](i, j) &= \mathbf{x}^v(i, j) + \Delta_v \cdot \delta_v[\mathbf{x}^v](i, j).\end{aligned}\tag{T3.5}$$

Note that,

$$\begin{aligned}A_q &= I + \Delta \cdot A_\delta \iff A_\delta = \Delta^{-1} \cdot (A_q - I_n); \\ B_q &= \Delta \cdot B \iff B_\delta = \Delta^{-1} \cdot B_q; \\ C_q &= C_\delta \iff C_\delta = C_q; \\ D_q &= D_\delta \iff D_\delta = D_q.\end{aligned}\tag{T3.6}$$

Here,  $\Delta = [\Delta_h I_{n_h} \oplus \Delta_v I_{n_v}]$  is of size  $(n_h + n_v) \times (n_h + n_v)$ .

The associated system theoretic notions, such as, transition matrix, transfer function, characteristic equation, etc., have also been introduced. This s.s. model is the basis for designing 2-D filters with superior finite wordlength properties. The design procedures developed are expected to be extremely useful in obtaining high- $Q$  2-D and  $m$ -D digital filters that are suitable for high-speed applications.

*Stability.* In the 1-D case, it has been shown that, direct techniques with no recourse to transformations (that first converts a given  $\delta$ -system to its  $q$ -system counterpart) can

provide numerically more reliable stability checking algorithms. With this in mind, for the 2-D case, a direct stability checking technique applicable to the corresponding  $\delta$ -system transfer function has been introduced. For this purpose, a recently developed tabular form was extended to the complex coefficient case and the notion of Schur-Cohn minors was introduced to the  $\delta$ -operator case.

*Gramians and balanced realization.* The notions of reachability and observability gramians and balanced realization have been introduced for the  $\delta$ -operator case. In order to do this, first, the relationship between the gramians for the  $\delta$ - and  $q$ -operator cases, as defined in the literature, was established. The reachability and controllability gramians, that is,  $P$  and  $Q$ , respectively, for 1-D  $\delta$ -systems were found to satisfy

$$\begin{aligned} P &= \frac{1}{2\pi j} \oint_{\mathcal{T}_\delta} (cI - A_\delta)^{-1} B_\delta B_\delta^* (c^*I - A_\delta^*)^{-1} \frac{dc}{1 + \Delta c}; \\ Q &= \frac{1}{2\pi j} \oint_{\mathcal{T}_\delta} (c^*I - A_\delta^*)^{-1} C_\delta^* C_\delta (cI - A_\delta)^{-1} \frac{dc}{1 + \Delta c}, \end{aligned} \quad (\text{T3.7})$$

where  $\mathcal{T}_\delta$  is the stability boundary applicable for  $\delta$ -systems, that is,  $\mathcal{T}_\delta = \{c \in \mathfrak{F} : |c + 1/\Delta| = 1/|\Delta|\}$ . An extension of this is then used to define the 2-D gramians of  $\delta$ -systems represented in the Roesser model developed above.

For the important class of separable (that is, separable-in-denominator) systems, it is shown that these gramians may be computed through the solution of four Lyapunov equations. These notions and results are useful in many applications, such as, in extracting reduced order models of  $\delta$ -systems.

*Sensitivity.* Measures that indicate coefficient sensitivity of the  $\delta$ -models developed above have been introduced. Unlike what is available in literature, this development is applicable to the MIMO case as well. With these sensitivity measures as a guide, development of minimum sensitivity structures has been carried out. The connection with the corresponding balanced realizations has been pointed out.

*Roundoff noise.* With the use of a noise model that takes into account the roundoff error propagation in the s.s. model developed above, structures that minimize roundoff noise have been developed.

### **Publications: Work directly related to grants**

- [1] K. Premaratne and P.H. Bauer (1994). Limit cycles and asymptotic stability of delta-operator systems in fixed-point arithmetic. *Proceedings 1994 IEEE International Symposium on Circuits and Systems (ISCAS'94)*, London, UK, vol. 2, 461-464.

- [2] P.H. Bauer and K. Premaratne (1994). Fixed-point implementation of multi-dimensional delta-operator formulated discrete-time systems: Difficulties in convergence. *Proceedings of the 1994 IEEE SOUTHEASTCON*, Miami, FL, 26-29.
- [3] K. Premaratne and A.S. Boujarwah (1994). An algorithm for stability determination of two-dimensional delta-operator formulated discrete-time systems. *Multidimensional Systems and Signal Processing*, to appear.
- [4] K. Premaratne, J. Suarez, M.M. Ekanayake, and P.H. Bauer (1994). Two-dimensional delta-operator formulated discrete-time systems: State-space realization and its finite wordlength properties. *37th Midwest Symposium on Circuits and Systems*, Lafayette, LA, to be presented; *IEEE Transactions on Signal Processing*, in preparation.
- [5] E.C. Kulasekere, K. Premaratne, P.H. Bauer, and L.J. Leclerc (1994). An exhaustive search algorithm for checking limit cycle behavior of digital filters. *IEEE Transactions on Signal Processing*, in preparation.

*Note.* Contents of [1-2] are being prepared for publication in *IEEE Transactions on Signal Processing*.

**Publications: Other work where grants are acknowledged**

- [1] K. Premaratne and E.I. Jury (1994). Discrete-time positive-real lemma revisited: The discrete-time counterpart of the Kalman-Yakubovitch lemma. *IEEE Transactions on Circuits and Systems—I. Fundamental Theory and Applications*, to appear.
- [2] M.M. Ekanayake and K. Premaratne (1994). Two-channel IIR QMF filter banks with approximately linear-phase analysis and synthesis filters. *28th Annual Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, to be presented; *IEEE Transactions on Signal Processing*, in review.
- [3] K. Premaratne and M. Mansour (1994). Robust stability of time-variant discrete-time systems with bounded parameter perturbations. *IEEE Transactions on Circuits and Systems—I. Fundamental Theory and Applications*, in review.
- [4] S.A. Yost and P.H. Bauer (1994). Robust stability of multi-dimensional difference equations with shift-variant coefficients. *Multidimensional Systems and Signal Processing*, to appear.

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 03 August 1994	3. REPORT TYPE AND DATES COVERED Interim; 01 January---30 June, 1994
----------------------------------	----------------------------------	---

4. TITLE AND SUBTITLE High-speed fixed- and floating-point implementation of delta-operator formulated discrete-time systems	5. FUNDING NUMBERS Grant No: N00014-94-1-0454
---	---

6. AUTHOR(S) Kamal Premaratne	R&T Project Code: 3148508-01
----------------------------------	---------------------------------

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Department of Electrical and Computer Engineering University of Miami P.O. Box 248294 Coral Gables, FL 33124	8. PERFORMING ORGANIZATION REPORT NUMBER
---	--

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research (ONR) Code 251:JWK Ballston Tower One North Quincy Street Arlington, VA 22217-5660	10. SPONSORING / MONITORING AGENCY REPORT NUMBER
--	--

11. SUPPLEMENTARY NOTES  
This project is a collaborative effort with Professor Peter H. Bauer, Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556, who is the principal investigator of Grant No: N00014-94-1-0387; R&T Project Code: 3148509-01.

12a. DISTRIBUTION / AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
--	------------------------

13. ABSTRACT (Maximum 200 words) This report addresses the analysis and design of finite word-length implementations of linear time-invariant delta-operator formulated discrete-time systems and the development of a 2-D delta-operator state-space model. It is shown that, in fixed-point arithmetic, linear time-invariant systems implemented with delta-operator do not generally outperform their shift-operator counterparts; they always show unstable limit cycle behavior and convergence to incorrect equilibria independent of realization and sampling time. Coefficient sensitivity is still superior. With floating-point arithmetic, delta-operator implementations consistently perform better than their shift-operator counterparts. They show superior quantization noise and sensitivity properties. Zero convergence problem of the fixed-point case does not exist if the mantissa length is sufficiently large. Noting these attractive finite wordlength properties, the concept of delta-operator has been extended to the multi-dimensional case. A 2-D state-space model, the notions of gramians, and balanced realization have been introduced. As for the 1-D case, sensitivity and roundoff noise behavior was analyzed. Realization that show 'minimum' sensitivity are equivalent to balanced realizations. The problem of directly checking stability in the delta-domain has also been addressed.

14. SUBJECT TERMS Delta-operator implementation of discrete-time systems, finite wordlength effects, coefficient sensitivity, quantization noise, limit cycles, fixed- and floating-point arithmetic, two- and multi-dimensional digital filters, stability.	15. NUMBER OF PAGES 09
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL
---	--	---	----------------------------------